SAVEETHA SCHOOL OF ENGINEERING
SIMATS, CHENNAI-602105

CSA0982 - Programming in Java For Networking

MINI PROJECT

Smart Traffic Signal Optimization

Done By:
- Vignesh S
192325002

1. Introduction:

Purpose of the App: The SmartTrafficSignalApp is designed to optimize traffic signal operations at multiple intersections, ensuring smooth traffic flow and minimizing wait times.

Key Features:

- User input for initial traffic signal setup.
- Automated random value generation for subsequent sets.
- Real-time traffic signal visualization.
- Emergency mode to handle urgent situations.
- Comprehensive reporting after each set.

Technology Stack:

- Java for back end logic.
- JavaFX for GUI.
- Random number generation for traffic signal timings.

2. System Design:

Architecture Overview:

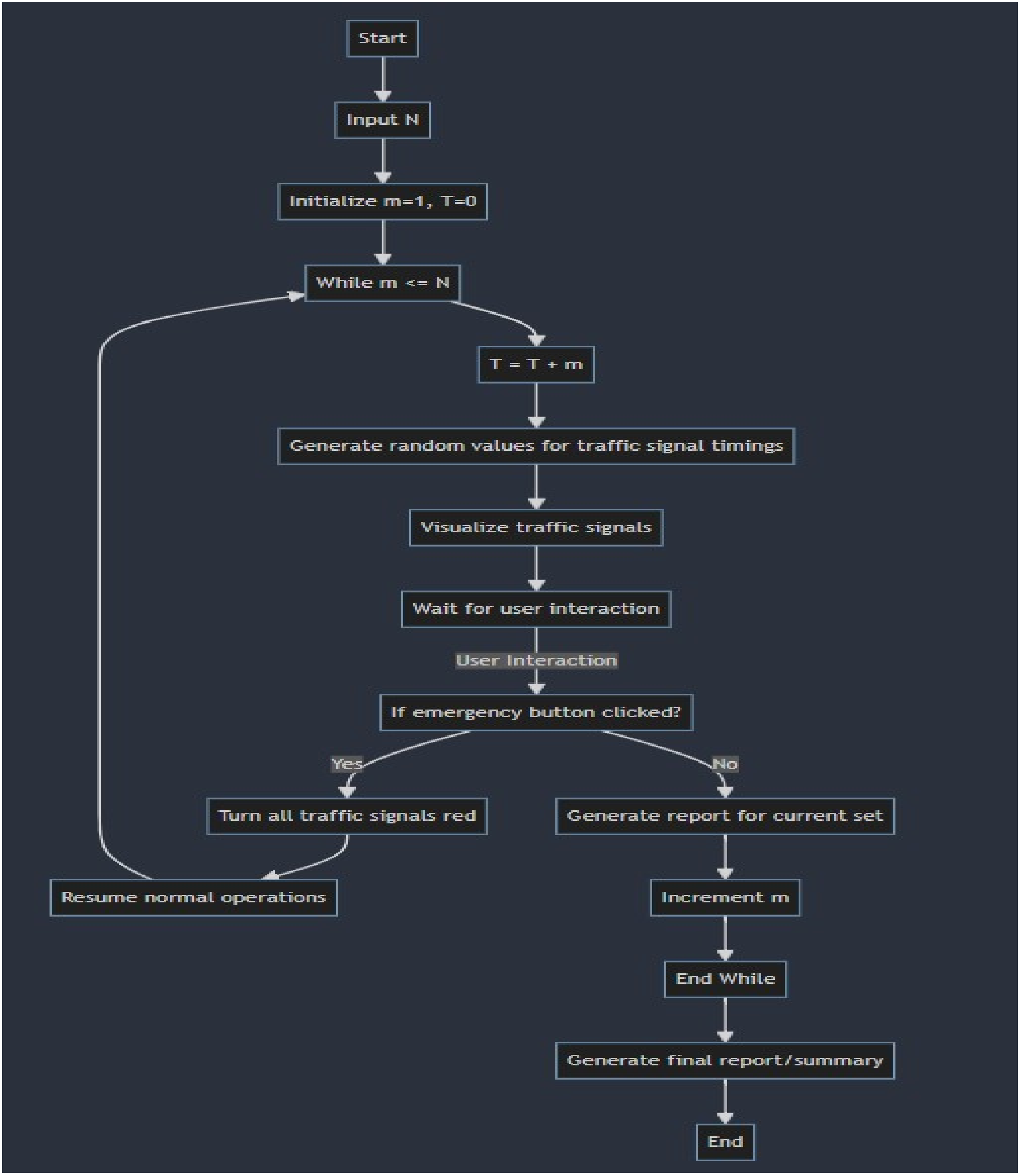
The app consists of the following main components:

- Traffic Signal Logic
- Random Value Generator
- User Input Handler
- Timer and Emergency Management
- JavaFX GUI

Components and Their Interactions:

- The TrafficSignal class handles the state of each traffic signal.
- The RandomValueGenerator class provides random timings for traffic signals.
- The UserInputHandler processes user input for the initial set.
- The TimerManager controls the timing for traffic signal changes.
- The EmergencyManager handles the emergency mode operations.

Data Flow Diagram:


Start
Input N
Initialize m=1, T=0
While m <= N
T = T + m
Generate random values for traffic signal timings
Visualize traffic signals
Wait for user interaction
User Interaction
If emergency button clicked?
Yes
No
Turn all traffic signals red
Generate report for current set
Resume normal operations
Increment m
End While
Generate final report/summary
End

3. Implementation Details

Traffic Signal Optimization Logic: The optimization logic ensures that traffic signals operate efficiently, minimizing wait times and avoiding overlapping signals. The app uses random value generation for signal timings, ensuring values are less than 250 and not repetitive for the same intersection.

Random Value Generation: The RandomValueGenerator method generates random values for traffic signal timings, ensuring they are less than 250 and not repetitive for the same intersection.

User Input Handling: The UserInputHandler class processes user input for the initial set of traffic signals. Subsequent sets are automated using the random value generator.

Timer and Emergency Handling: The TimerManager class controls the timing for traffic signal changes, while the EmergencyManager class handles emergency mode operations, ensuring all signals turn red during emergencies.

JavaFX GUI Components: The app uses various JavaFX components for the GUI, including buttons for starting the optimization process and handling emergencies, panes for visualizing traffic signals, and labels for displaying status messages.

Key Classes and Methods:

- TrafficSignal: Manages the state of each traffic signal.
- TrafficLightPanel: Determines the traffic light to be given for the input data.
- TrafficData: Receives data from the user.
- TrafficSignalOptimizer: Contains methods and functions that handles the signal timings and conditions.
- RandomValueGenerator: Generates random timings for traffic signals.
- UserInputHandler: Processes user input for the initial set.
- TimerManager: Controls the timing for traffic signal changes.
- EmergencyManager: Handles emergency mode operations.

4. Visualization and Reporting:

I) Real-time traffic conditions and signal timings.
II) Traffic flow improvements over time.
III) Average wait times at intersections.
IV) Overall congestion reduction.

5. User Interaction:

- ➢ Interface to monitor and manually adjust signal timings if needed.
- ➢ Dashboard to view performance metrics and historical data.
- ➢ The user interface is designed to be intuitive and informative, allowing users to interact with the system effectively.

6. Testing:

Testing Scenarios:

- Validate functionality under various traffic conditions.
- Test edge cases and error handling.
- Comprehensive test cases to ensure correct signal timing adjustments.

Results:

□ The system successfully adjusts signal timings based on real-time data.

□ Improvements in traffic flow and reduction in congestion were observed in simulated tests.

7. Conclusion:

The Smart Traffic Signal Optimization system effectively utilizes real-time data to dynamically adjust traffic signal timings, improving traffic flow and reducing congestion in urban areas. The system's design, implementation, and testing demonstrate its potential to enhance traffic management in smart cities.

8. Future Improvements:

- Incorporate machine learning algorithms to predict traffic patterns.
- Integrate additional data sources such as weather and event data.
- Expand the system to cover more intersections and incorporate multi-modal transportation data.